How to Use this Template

1. Make a copy [File → Make a copy...]
2. Rename this file: "**Capstone_Stage1**"
3. Replace the text in green

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"

Description
Intended User
Features
User Interface Mocks
    Screen 1
    Screen 2
Key Considerations
        How will your app handle data persistence?
        Describe any corner cases in the UX.
        Describe any libraries you'll be using and share your reasoning for including them.
Next Steps: Required Tasks
    Task 1: Project Setup
    Task 2: Implement UI for Each Activity and Fragment
    Task 3: Your Next Task
    Task 4: Your Next Task
    Task 5: Your Next Task

GitHub Username: kittcoldfire

Simple Audiobook Player

Description

Audiobook Player is a sleek, well designed Audiobook Player made to take all the worry out of listening to your favorite Audiobooks. It automatically keeps track of all your audiobooks, their playback positions and allows you to easily modify your listening experience. Gives you the ability to speed up the dialog, set a simple sleep timer for automatic playback halting and choosing album art from online sources.

Intended User

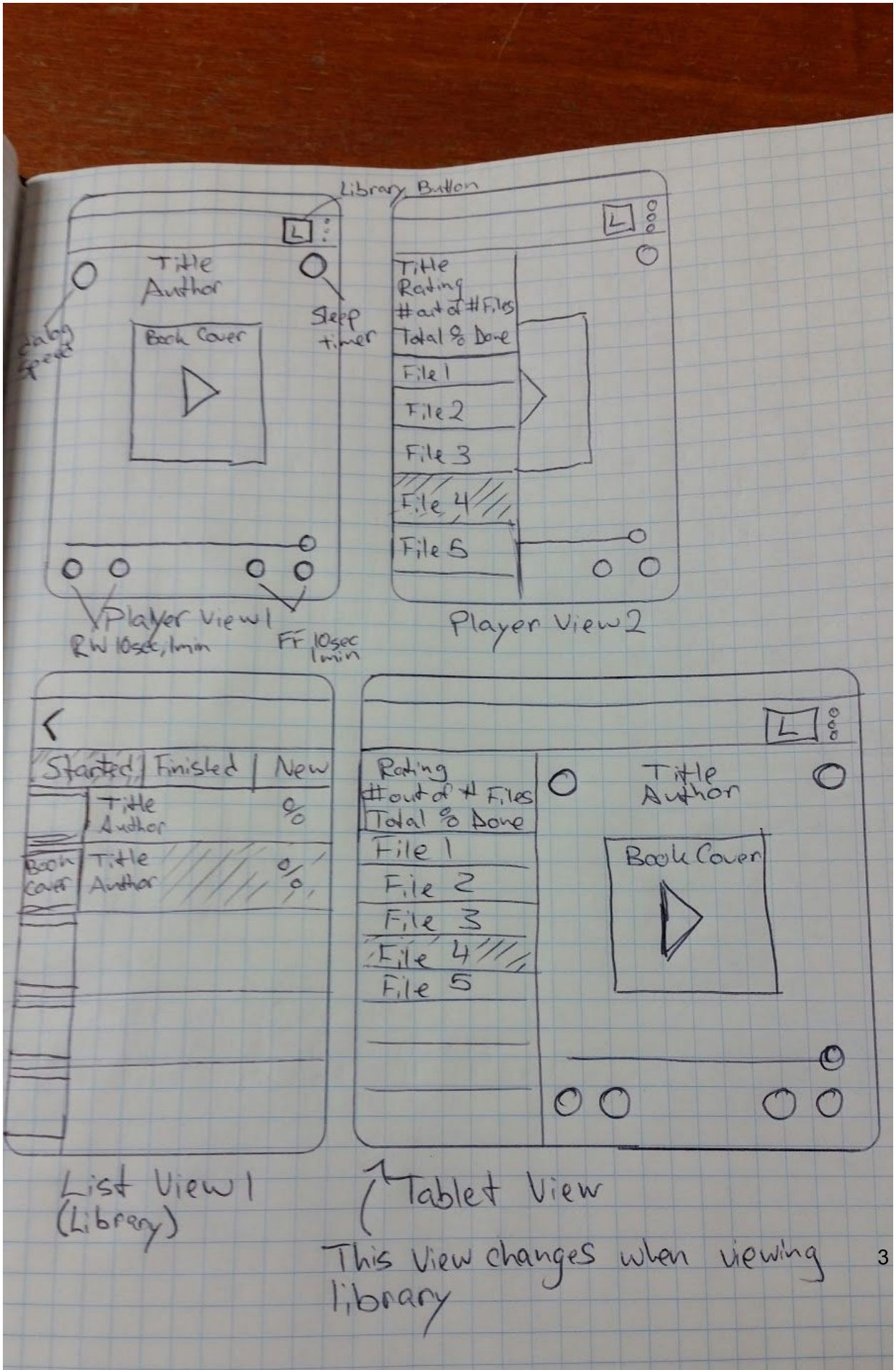
Someone who listens to audiobooks and wants a simple clean player to keep track of all of them.

Features

- Saves all playback positions
- Download cover art
- Choose dialog speed
- Configurable sleep timer
- Rate the books you've listened too
- Keep track of all books listened to (without storing the files on your device)

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.


Library Button
Title
Author
Book Cover
Sleep timer
Player View1
RW 10sec, 1min
FF 10sec 1min
Title
Rating
out of # Files
Total % Done
File 1
File 2
File 3
File 4
File 5
Player View2
Started | Finished | New
Title
Author
%
Book cover
List View1
(Library)
Tablet View
This View changes when viewing library

Key Considerations

How will your app handle data persistence?

I will copy the needed information from the MediaStore database already established on Android into my own database, and add columns that I will need to track as well (last playback position, history of listening etc).

Describe any corner cases in the UX.

Hitting back from the playlists brings the user back to the player screen. Hitting back on the player screen closes the Activity view but the MediaPlayer service continues playing (if it was playing).

Describe any libraries you'll be using and share your reasoning for including them.

I will use Glide to download and cache cover art for the books. Also will be connecting to the Google Books api for downloading information and cover art of the books.
No Nonsense File Picker Library to select the folder that the audio books are stored in (possibly also select files directly if you don't want to put in specific folders)
https://github.com/spacecowboy/NoNonsense-FilePicker

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Setup and configure your gradle build file to use the Glide library and File Picker.
- compile 'com.nononsenseapps:filepicker:2.5.0'
- compile 'com.github.bumptech.glide:glide:3.7.0'
- compile 'com.android.support:support-v4:19.1.0'

Task 2: Implement UI for Each Activity and Fragment

- Build UI for Player Interface
- Build UI for list items

Task 3: Create and Implement Database/Content Provider

- Create schema for our own database (and needed columns from MediaStore)
- Make the necessary files to copy data from the MediaStore database into our database.
- Create a listener to listen for when Media changes so we can update our database if need be, this can be easy as we don't need to delete files from our database, only added (and only if added to the folder we're watching)

Task 4: Create Media Player Service

- Create our service that will be our Media Player Session

Task 5: Hook Data and Views Together

- Create the link between views and data in our database.
- Hook in the service to on screen controls.

Task 6: Update Database on Playback

- Hook our player into our Content Provider to update our database when needed

Task 7: Implement Notifications

- Extend the service to use notifications on lock screens, notification bar and Android Wear devices.

Add as many tasks as you need to complete your app.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"